Rapport de stage

Programmation d’un logiciel de pilotage et de traitement de données pour un banc de caractérisation électrique des vibrations d’une micropoutre

Thomas HAËCK - Master 1 TdSI

Encadrant : Étienne LEMAIRE

Stage du 18/06/12 au 07/09/12

Résumé :

Dans le cadre de la thèse d'Étienne Lemaire, la caractérisation de propriétés viscoélastiques d'un fluide s'effectue via un capteur constitué d'une micro-poutre en silicium immergée dans le fluide. Un courant est envoyé dans la micropoutre, plongée dans un champ électromagnétique constant. Cette micropoutre est assimilée à un système résonant du second ordre. Il faut ensuite mesurer l'amplitude et la phase de son mouvement, afin d'en déduire des caractéristiques telles que la fréquence de résonance, la fréquence propre et le facteur de qualité, qui détermineront les caractéristiques viscoélastiques du fluide. Pour l'instant, cette mesure est effectuée de manière optique via un vibromètre.

À terme, les mesures devront être réalisées de manière électrique, via une piézorésistance mesurant la contrainte à l'encastrement. Pour ce faire, une carte électronique d'amplification et d'interface a déjà été réalisée. Les mesures se font actuellement via un analyseur Gain/Phase. Pour des raisons de coût, d'encombrement et de simplification, il serait idéal de réaliser le pilotage et l'acquisition des données via l'entrée/sortie d'une carte son d'ordinateur. Le but de ce stage est donc de concevoir et tester un programme Matlab permettant d'une part l'envoi d'un signal d'excitation vers la micropoutre, et d'autre part la réception des mesures. Ces dernières étant effectuées dans un environnement très bruité, il sera nécessaire de leur appliquer un traitement pour les rendre exploitables. Le programme doit aussi permettre, de la manière la plus automatique qui soit, de déterminer la fréquence de résonance, la fréquence propre et le facteur de qualité du capteur dans un fluide donné.

INDEX

INTRODUCTION

J'ai effectué mon stage de Master 1 en Traitement des Signaux et de l'Image du 18 Juin au 8 Septembre 2012 au sein du laboratoire de l'Intégration du Matériau au Système (IMS) à Talence. L'IMS est un laboratoire qui se positionne dans le domaine des sciences de l'ingénierie et des systèmes, en offrant une passerelle entre les Sciences et Technologies de l'Information et de la Communication et les Sciences Pour l'Ingénieur.
Dans ce laboratoire, j'ai été intégré plus précisément dans l'équipe MMM du groupe Microsystèmes, dirigée par Isabelle Dufour. J'ai été encadré par Étienne Lemaire, doctorant à l'IMS, encadré lui-même par Benjamin Caillard. L'équipe MMM se concentre sur le développement de capteurs à base de microleviers. Ils sont utilisés comme capteurs de gaz, d'humidité, de viscosité, etc...

L'utilisation qui nous intéresse dans le cadre de ce stage est le capteur de viscosité et de densité. En effet, comme nous le verrons, les caractéristiques d'une micropoutre en tant qu'oscillateur résonant varient en fonction du milieu dans lequel elle est plongée.
Nous nous intéresserons ensuite plus au fonctionnement du capteur, à sa chaîne d'actionnement et à sa chaîne de mesure, comprenant les différentes cartes d'acquisition et de pilotage ayant déjà été développées par le Laboratoire d'Analyse et d'Architecture des Systèmes (LAAS), et nous verrons comment intégrer un ordinateur muni d'une carte son dans ce banc de mesure.
Puis, nous verrons les différentes méthodes qui ont été envisagées, afin d'obtenir les propriétés de la micropoutre telles que la fréquence de résonance, la fréquence propre et le facteur de qualité dans un milieu donné.

I - Notions de rhéologie

<u>Définition</u> : la rhéologie (du grec *rheo*, couler et *logos*, étude) est l'étude de la déformation et de l'écoulement de la matière sous l'effet d'une contrainte appliquée.

Les mesures rhéologiques peuvent par exemple être utilisées dans le cadre de la formulation de différents produits industriels, comme les huiles, les bitumes ou des produits agroalimentaires (yaourts).
Les propriétés de viscoélasticité et de densité d'un fluide permettant de décrire son écoulement, leur étude rentre directement dans le cadre de la rhéologie. Considérons un fluide placé entre deux plans horizontaux, et appliquons une force tangentielle, dite de cisaillement, sur le plan supérieur.

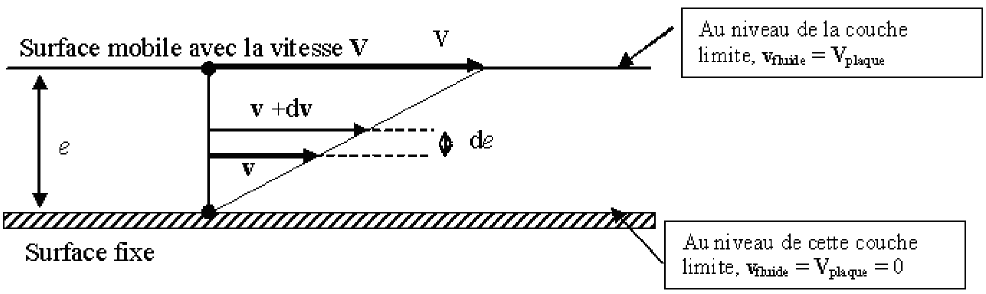


Figure 1 : mouvement de cisaillement [1]

On peut ici considérer le fluide comme un empilement de couches, qui vont glisser les unes par rapport aux autres sous l'effet de la force de cisaillement. Chaque couche se déplacera dans la direction de la force appliquée, les couches inférieures se déplaçant moins vite. Ce "retard" est dû à la viscosité du fluide. Lorsque la viscosité est indépendante de la vitesse de cisaillement $\frac{dv}{de}$, le fluide est dit newtonien. Sinon, le fluide sera dit non newtonien.

Un fluide visqueux peut être modélisé par un amortisseur et un ressort en série, d'après le modèle de Kelvin-Voigt :

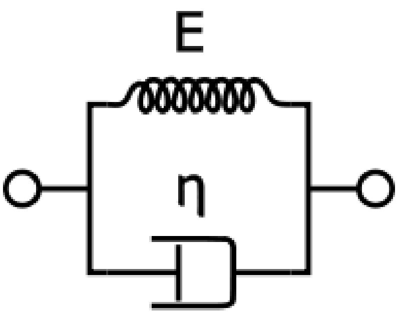


Figure 2 : modèle de Kelvin-Voigt

Il existe plusieurs types de capteurs, appelés rhéomètres, mesurant les caractéristiques rhéologiques d'un fluide. Les plus fréquents sont des rhéomètres rotatifs (cylindriques, cône-plan, plan-plan) :

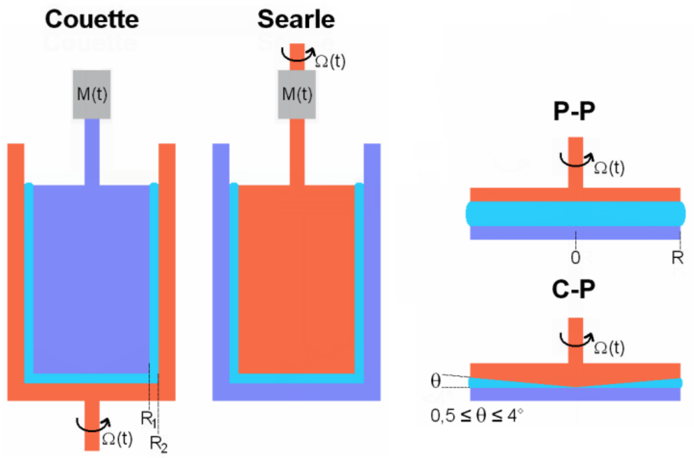


Figure 3 : Différents rhéomètres [2]

Il est aussi possible de mesurer la viscosité d'un fluide en mesurant le temps que met une bille à chuter dans une hauteur connue du fluide.

L'étude des propriétés rhéologiques peut aussi s'effectuer à l'échelle nanométrique, en étudiant le déplacement de particules nanométriques bien connues dans le fluide.

Enfin, dans le cadre qui nous intéresse, on peut utiliser une poutre de dimension micrométrique plongée dans le fluide, que l'on va actionner sur une large gamme de fréquences. Cette poutre constituant un oscillateur, le spectre de résonance mécanique de la micropoutre sera dépendant du fluide dans lequel elle est immergée.

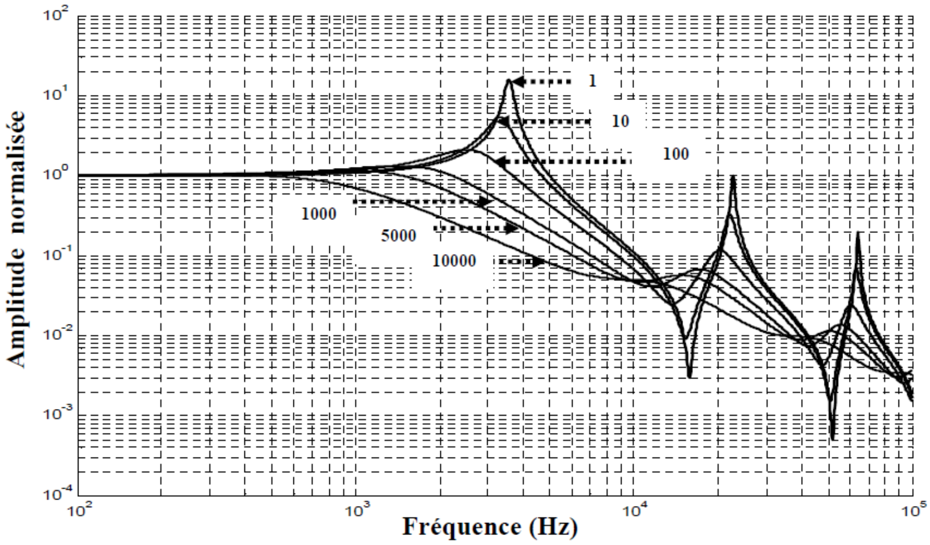


Figure 4 : Spectre simulé de résonance mécanique du microlevier (50x400x3000µm) immergé dans des huiles silicones de viscosité η=1cP et de masses volumiques respectives : ρ= 500, 1000, 5000, 10 000 kg.m-3 [3]

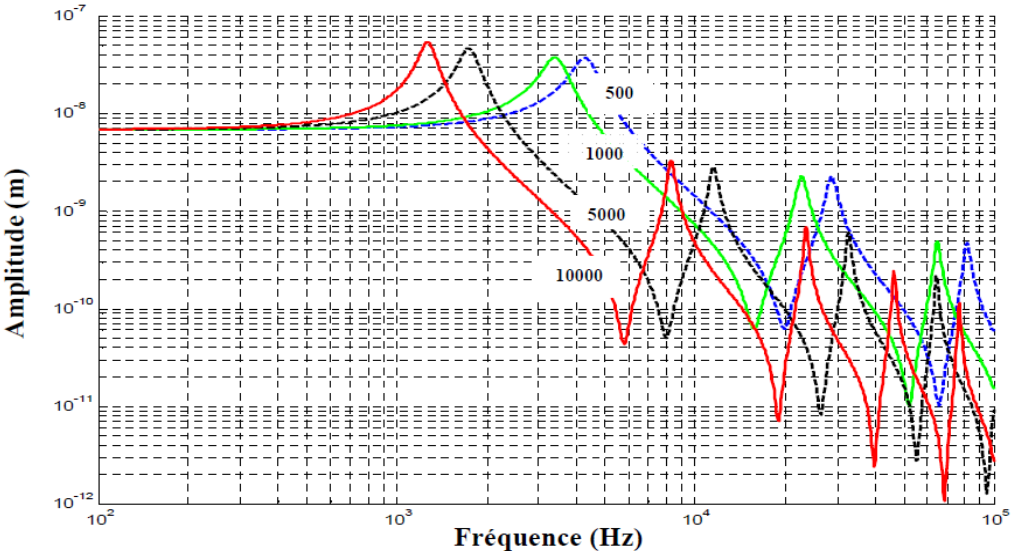


Figure 5 : Spectre simulé de résonance mécanique du microlevier (50x400x3000µm) immergé dans des huiles silicones de viscosité η=1cP et de masses volumiques respectives : ρ= 500, 1000, 5000, 10 000 kg.m-3 [3]

On peut constater, d'après la figure 2, que l'augmentation de la viscosité entraîne une forte augmentation de l'amortissement et réduit légèrement la fréquence de résonance. D'après la figure 3, l'augmentation de la masse volumique diminue fortement la fréquence de résonance.

Les avantages principaux des capteurs à base de micropoutres sont :
- leur faible encombrement
- leur faible consommation électrique
- leur coût relativement faible
- un temps de réponse assez rapide

II - Banc de mesure

1 - Description de la micropoutre

Cette poutre, fabriquée en SOI (Silicon on Insulator), possède une piste électrique en or qui suit les contours de la surface supérieure. Elle est placée dans un champ magnétique fixe (aimant). On envoie un courant d'actionnement I sur la piste.

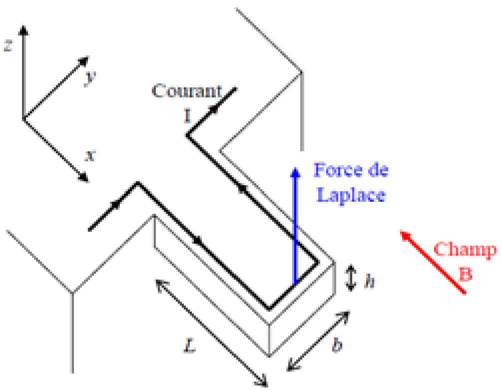


Figure 6 : actionnement de la micropoutre [1]

La présence d'un courant dans un champ magnétique entraîne l'apparition de forces de Laplace ($F = I \wedge B$). Sur les côtés de la poutre, le champ étant colinéaire à la piste, la force de Laplace est nulle. En bout de poutre, il y a apparition d'une force de Laplace orthogonale au plan de la poutre. La poutre subira donc une déformation. Si le courant appliqué est alternatif, la poutre oscille.

Il s'agit ensuite de mesurer ces oscillations. Ces mesures s'effectuent actuellement de manière interférométrique grâce à un vibromètre. Cet instrument est malheureusement assez volumineux et très onéreux. Il présente aussi l'inconvénient majeur de ne pas pouvoir être utilisable dans le cas de fluides opaques.

Une autre manière de mesurer les oscillations est de placer une piézorésistance à l'encastrement de la micropoutre où la contrainte mécanique est maximum. La variation de résistance dR est directement proportionnelle à la déformation subie, et donc à l'oscillation de la poutre. Pour mesurer cette variation, on utilise un montage semblable à un pont de Wheatstone (mesure différentielle).

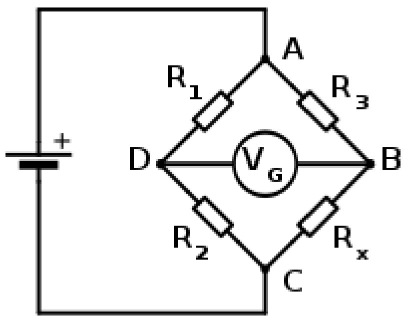


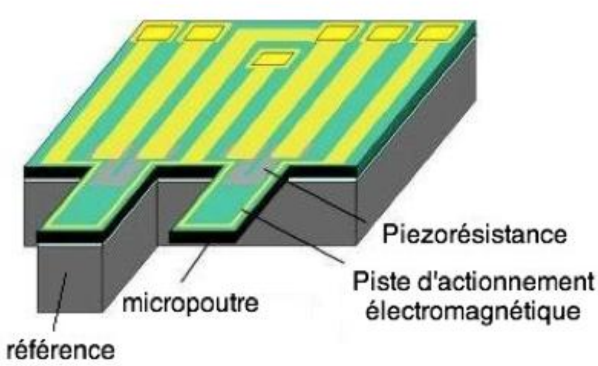
Figure 7 : pont de Wheatstone - Rx est la piézorésistance

Ce pont est alimenté par une tension continue, la mesure s'effectuant entre les points B et D. Lorsque R_1=R_2=R_3=R_x, le pont est à l'équilibre et V_G = 0V.

Si R_1=R_2=R_3=R et que R_x varie faiblement (R_x = R + dR), on a $V_G = \frac{dR}{4R} V_{AC}$

La variation de la piézorésistance entraîne donc une variation de tension, qui est facilement mesurable.

Afin d'obtenir une mesure plus précise et de compenser les différents bruits, une seconde micropoutre, fixe, est placée à côté de la micropoutre mobile. Elle est dotée elle aussi d'une piste d'actionnement, ainsi que d'une piézorésistance et d'une piste de mesure, et est donc soumise au même milieu. Les phénomènes parasites intervenant sur les deux poutres à la fois, en plaçant la piézorésistance en R_3 sur le schéma précédent, on viendra compenser une partie des bruits de mesure.



Les pistes d'actionnement et de mesure étant assez proches, il y a présence d'un phénomène de couplage dépendant du milieu dans lequel la poutre est plongée. Grâce à cette poutre fixe, ce couplage a pu être précédemment étudié et caractérisé.

2 - Électronique de commande et de mesure

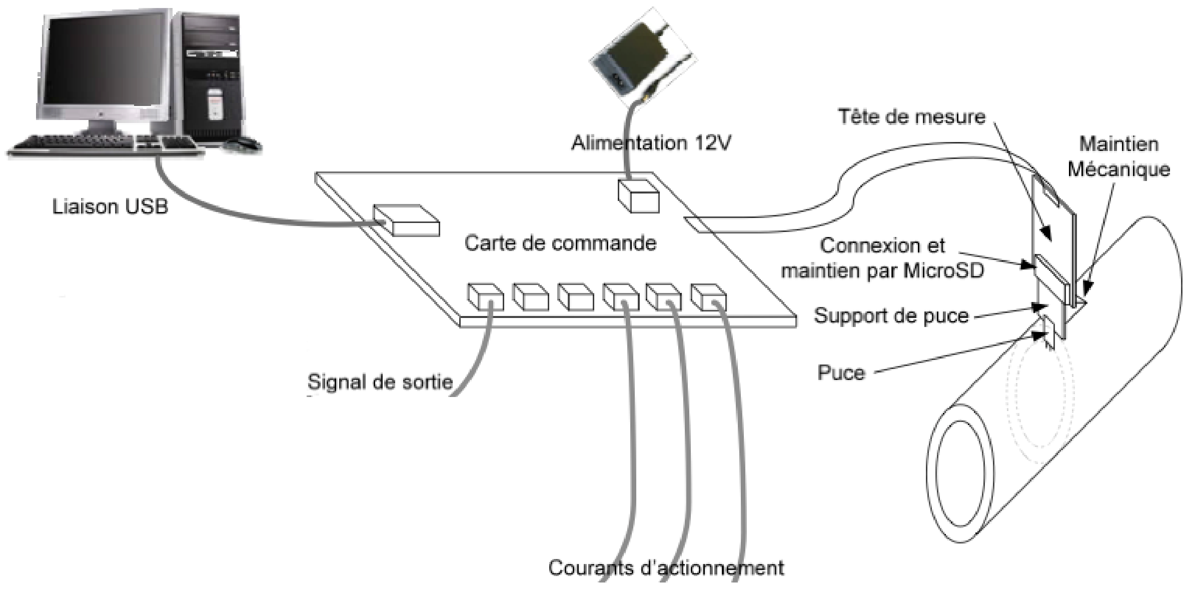


Figure 8 : système de détection et de commande [4]

a) Support de puce

Il est utilisé pour fixer la micropoutre au banc de mesure et sert d'intermédiaire de connexion entre la micropoutre et la tête de détection.

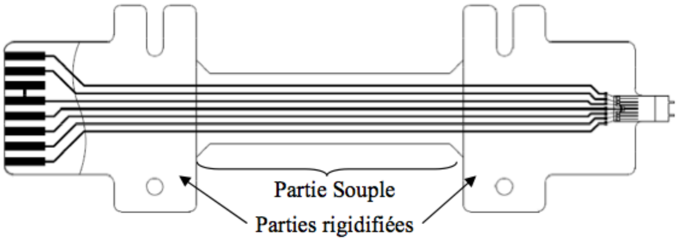


Figure 9 : support de puce

b) Tête de détection

Comportant uniquement de l'électronique analogique, cette tête de détection permet de générer les courants de polarisation et comporte un premier étage d'amplification différentielle.

c) Carte de commande

Cette carte comporte un deuxième étage d'amplification, et permet de compenser les erreurs de mesures dues à la dispersion des piézorésistances et les offsets générés par les cartes précédentes.

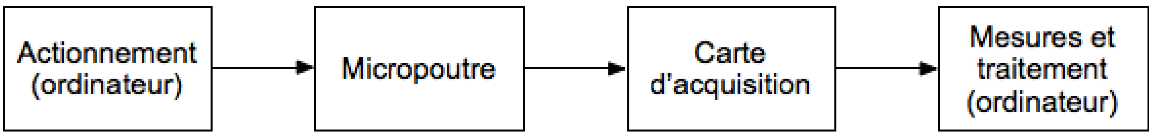
La compensation est réalisée à l'aide d'un microcontrôleur, d'un convertisseur analogique-numérique 16 bits et d'un convertisseur numérique-analogique 16 bits, le tout contrôlé via bus USB par un ordinateur via une interface utilisateur.

d) Ordinateur

Outre le fait qu'il soit nécessaire pour la compensation, un ordinateur peut être utilisé pour générer les courants d'actionnement et récupérer le signal de sortie.

En effet, les micropoutres que nous utilisons ont une fréquence de résonance dans l'air comprise entre 0 et 20 kHz. Le spectre de résonance que nous voulons récupérer se situe donc dans la gamme de fréquence audible par l'homme, gamme pour laquelle ont été conçues les cartes son d'ordinateur.

On pourra donc générer les courants d'actionnement via la sortie haut-parleurs de l'ordinateur, et récupérer le signal de mesure par l'entrée microphone.



Nous avons donc maintenant entièrement décrit la chaîne de mesure et d'actionnement de notre expérimentation. Il s'agit maintenant de savoir exactement quel signal d'actionnement il nous faut générer, et comment traiter les mesures obtenues.

III - Méthodes de mesure du gain et de la phase

1 - Actionnement et acquisition

L'envoi (actionnement) et la réception (acquisition) de signaux audio se fait directement sous Matlab par le biais de fonctions simples (audioplayer, audiorecorder). Ces fonctions ont seulement besoin des paramètres physiques de la carte son que sont la fréquence d'échantillonnage (44100 Hz) et le nombre de bits du convertisseur analogique-numérique (16 bits).

2 - Mesures mono

Dans le cadre de la première partie de mon stage, la connectique présente ne permettait pas d'exploiter la possibilité d'avoir 2 signaux de sortie et 2 signaux d'entrée. Il m'a donc fallu développer des méthodes de mesure en conséquence.

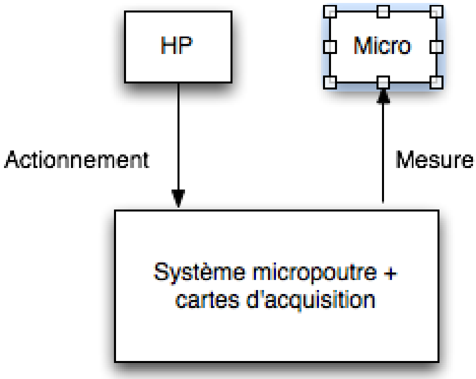


Figure 10 : schéma des branchements mono

a) Mesure de gain : moyenne de transformées de Fourier

On envoie ici directement un signal d'excitation (sweep) d'une durée d'une seconde, dont la fréquence varie linéairement entre 100 Hz et 22 kHz et dont l'amplitude est constante.
On prend ensuite la transformée de Fourier du signal mesuré au niveau du haut-parleur, dont on calcule le module.
Ceci permet d'obtenir une première mesure du spectre de gain de la micropoutre, entre 100 Hz et 22 kHz. Étant donné que le signal est très bruité, une seule mesure n'est pas suffisante. On prend donc N mesures consécutives, et on calcule la moyenne des modules des transformées de Fourier.

Le problème de ce type de mesure est qu'on n'obtient aucune information de phase. En effet, le déphasage mesuré entre le signal d'excitation et le signal mesuré n'est pas le déphasage de la micropoutre. Tous les autres systèmes entrant en jeu (haut-parleur, cartes d'acquisition, microphone, matlab) entraînent chacun un déphasage, et rien ne nous dit que ce déphasage est constant en fonction de la fréquence.

b) Mesure de la fréquence propre par Zero-Crossing

Dans cette méthode, on envoie toujours un sweep dont la fréquence varie entre 100 Hz et 22 kHz. Sur le signal d'entrée, on note les indices où le signal passe à zéro grâce à une fonction zero_crossing. On fait de même pour le signal mesuré. Théoriquement, si le système était parfait, on pourrait retrouver facilement le déphasage pour chaque fréquence à partir du vecteur Δt, qui représente le vecteur de différences entre les temps de passage à zéro de la mesure et les temps de passage à zéro de l'entrée.

Le déphasage φ vaut $\varphi(f) = \Delta t * 2\pi * f$, avec fs la fréquence d'échantillonnage.

Le problème est que nous avons un déphasage inconnu entre l'entrée et la sortie du système, déphasage qui n'est pas forcément dû qu'à la micropoutre. L'idée a donc été de regarder ce que donnait φ, et de voir si cela correspondait au portrait de phase d'un système oscillatoire du second ordre : phase constante au début, puis une chute importante, suivie d'une stabilisation. Dans le cas contraire, on décale vers le bas les indices de zero-crossing de la mesure, et ainsi de suite jusqu'à avoir un profil de phase ressemblant au profil théorique.

Ensuite, en faisant une simple dérivée, on obtient le moment où la phase a la pente vers le bas la plus forte, ce qui correspond au moment où $f = f_0$, la fréquence propre du système.

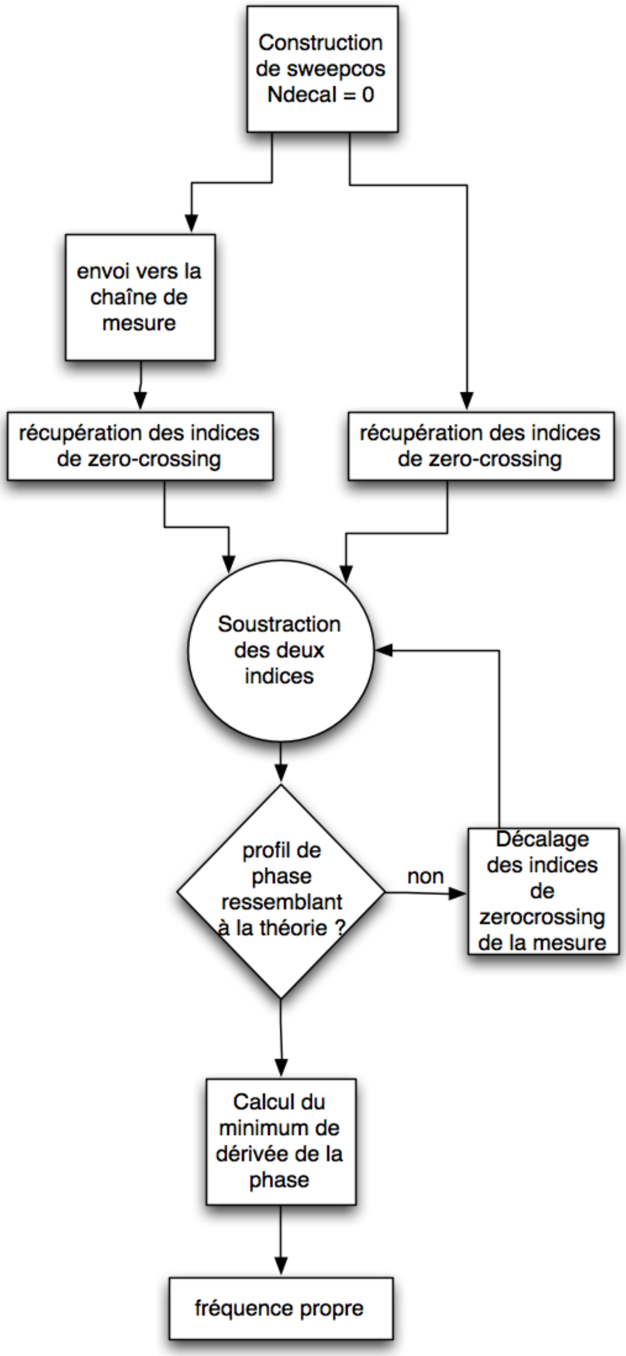


Figure 11 : schéma-bloc de principe du zero-crossing

3 - Mesures stéréo

Une fois la connectique obtenue, il a été possible de se servir des deux voies des haut-parleurs et du micro. La voie 1 du micro a donc été bouclée sur la voie 1 des haut-parleurs :

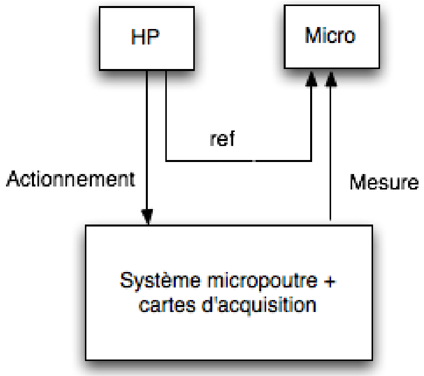


Figure 12 : schéma des branchements stéréo

La partie précédente concernant la mesure du gain par moyenne de module des transformées de Fourier a été adaptée pour permettre à la fois une mesure du spectre de la poutre et une mesure du spectre de la référence.

a) Détection synchrone

La détection synchrone consiste à envoyer un signal sinusoïdal à une pulsation fixe ω vers le capteur. En sortie du capteur, on récupère un signal A.cos(ωt) +b(t). Ce signal de mesure est ensuite multiplié par le signal d'excitation, cos(ωt).

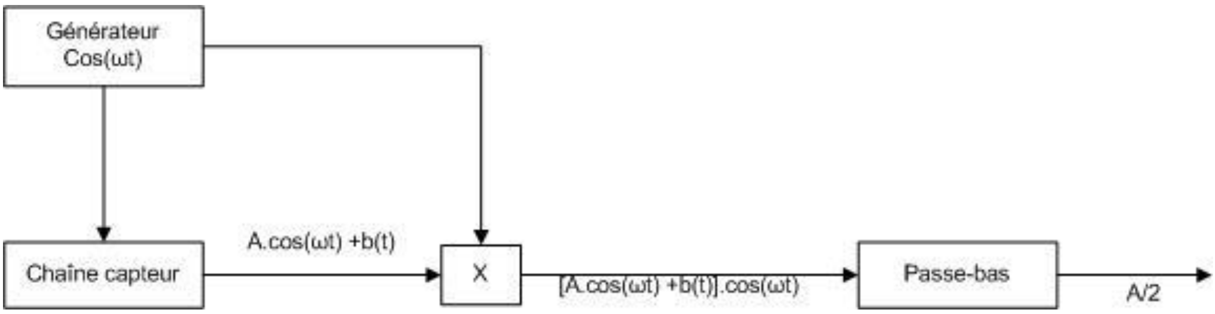


Figure 13 : détection synchrone simple

En sortie du multiplieur, on obtient donc [A.cos(ωt) +b(t)].cos(ωt) = A/2 + A/2 cos(2ωt) + b(t)cos(ωt), où l'on considérera que b(t) est un bruit blanc.
On se rend compte qu'il y a une composante continue dans ce signal, qui correspond au gain A du capteur à la pulsation ω. En appliquant un filtre passe-bas au signal, on retrouve donc A/2 en sortie.

Ici nous n'avons pas un signal du type A.cos(ωt) +b(t) en sortie du capteur, mais un signal A.cos(ωt+ϕ) (présence d'un déphasage). Si l'on effectue seulement une détection synchrone simple, on aura donc A/2 cosϕ en sortie du passe-bas soit deux informations mélangé dans un signal. La double détection synchrone permet de pallier à ce problème et d'extraire d'une part l'amplitude (A) et d'autre part la phase (ϕ).

b) Double détection synchrone

Le principe de la double détection reste le même que pour la détection synchrone simple. On envoie un signal cos(ωt) à l'entrée du capteur, et on récupère A.cos(ωt+ϕ) en sortie. La branche supérieure du schéma de double détection synchrone est la même que celle de détection synchrone simple.

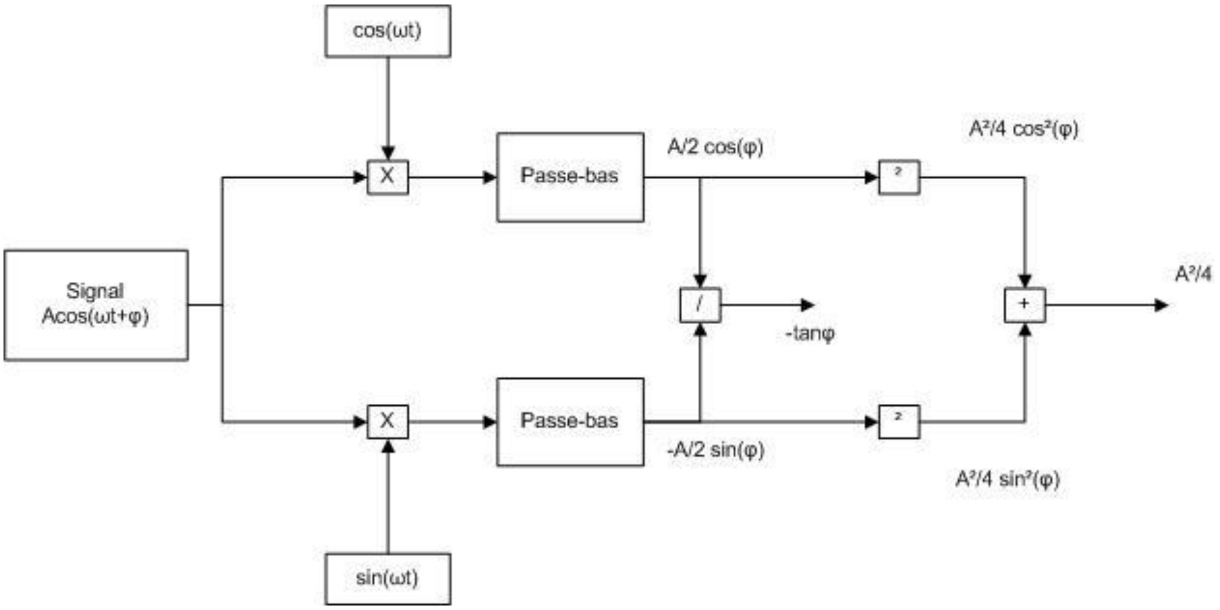


Figure 14 : double détection synchrone

Sur la branche inférieure, on multiplie le signal de sortie du capteur par un sinus, puis on applique le même filtre passe-bas sur les deux branches. On obtient donc A/2 cosϕ et - A/2 sinϕ. Via de simples opérations arithmétiques, il est ensuite facile de remonter au gain A et à la phase ϕ du système, à la pulsation ω. Mais nous n'avons pas encore le spectre entier.

c) Discrétisation

En effet, la méthode précédente permet seulement d'obtenir le gain et le déphasage pour une fréquence donnée. Pour obtenir les spectres de gain et de phase sur la totalité de la plage de fréquences (entre 100 Hz et 22 kHz), nous aurions besoin, comme dans la première partie, d'envoyer un sweep en entrée du capteur.

Malheureusement, le sweep présente une variation continue de fréquence au cours du temps, il ne peut pas être applicable directement dans la double détection synchrone. J'ai donc décidé de réaliser un sweep discrétisé.

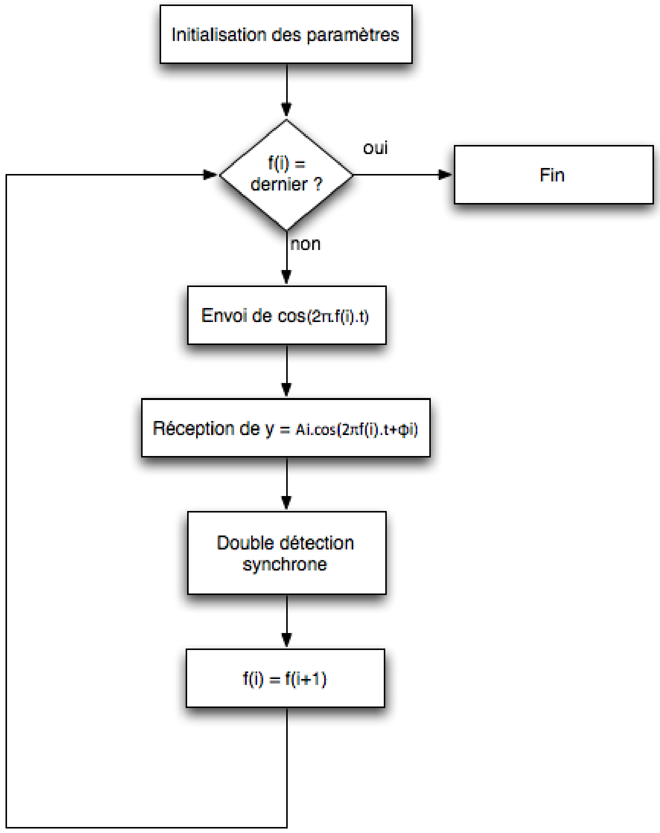


Figure 15 : schéma-bloc de la double détection synchrone discrète

Concrètement, on réalise un tableau de fréquences balayant le spectre total (100 Hz à 22 kHz). Le signal de sweep discrétisé sera constitué de sinusoïdes de chacune des fréquences du tableau mises bout à bout. En somme, on réalise une double détection synchrone à la fréquence f_1, puis une autre à la fréquence f_2, et ainsi de suite.

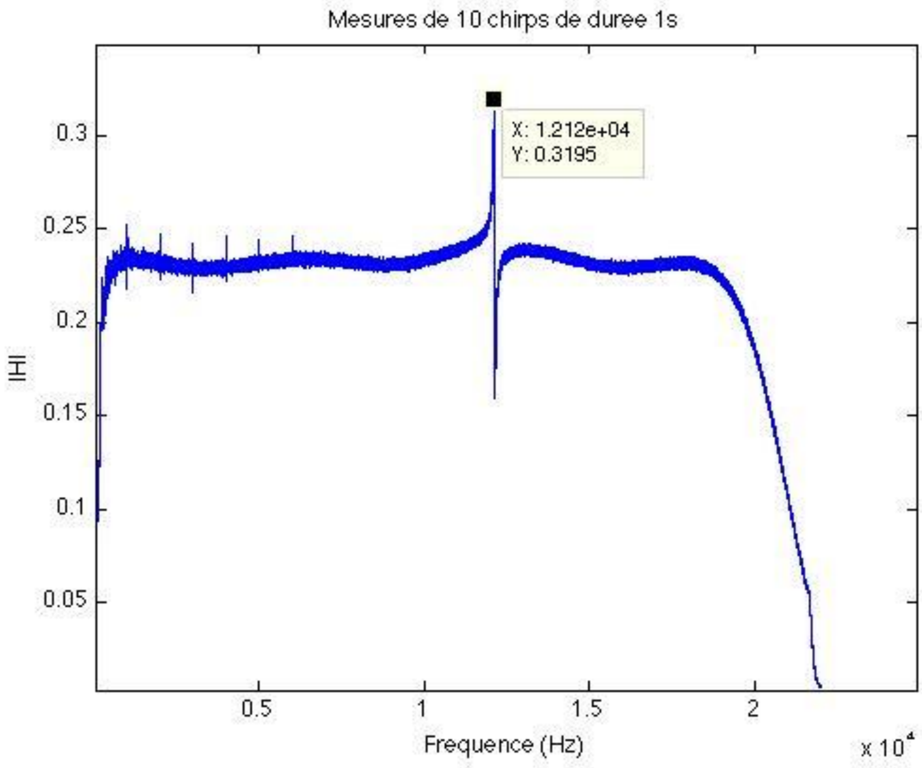
La fonction réalise tout d'abord un balayage en fréquence rapide sur tout le spectre (100 Hz à 22 kHz), puis récupère la fréquence pour laquelle le gain est maximal, et qui a donc le plus de chances d'être proche de la fréquence de résonance.

Elle effectue ensuite un deuxième balayage fréquentiel, plus fin, autour de la fréquence maximale obtenue plus haut, ce qui permet d'obtenir un profil précis du gain au voisinage de la fréquence de résonance.

IV Résultats de mesures

1 - Moyenne de module de transformées de Fourier :

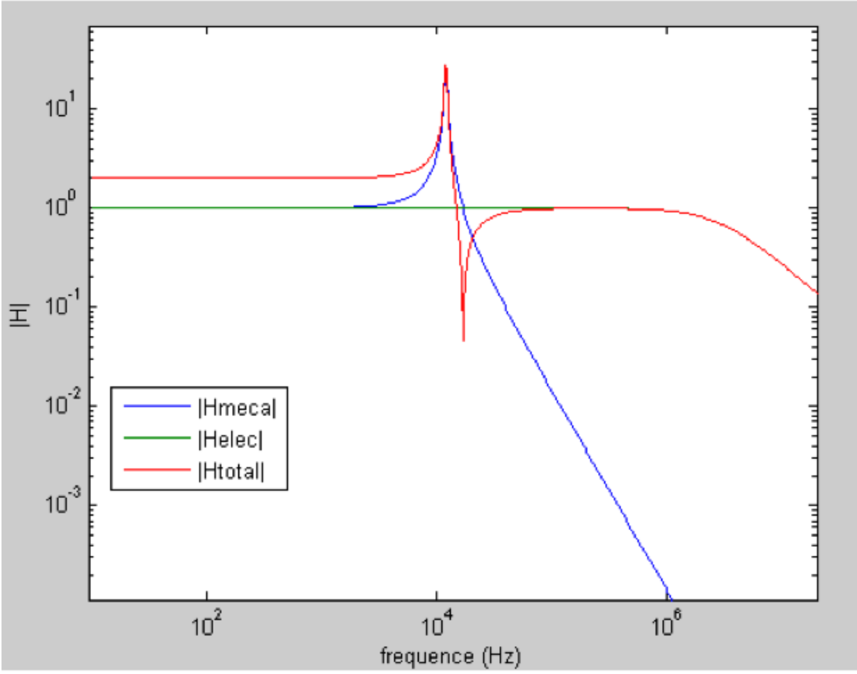
Voici les mesures de gain obtenues en réalisant une moyenne sur 10 acquisitions de 1s du module de la transformée de Fourier du signal reçu. Elles ont été effectuées avec une micropoutre de type LH dans l'air.



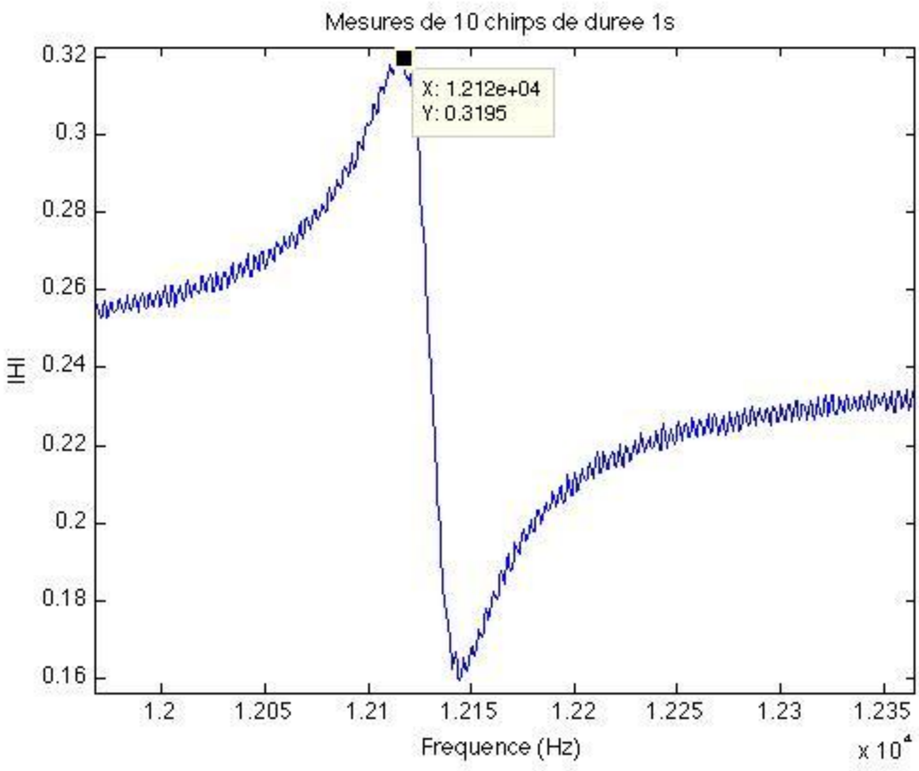
La fréquence de résonance f_r se remarque très rapidement, on a f_r = 12 120 Hz. Ce résultat est cohérent avec les résultats expérimentaux obtenus avec la mesure optique (vibromètre).

La pente observée dans les hautes fréquences est due à la présence d'un filtre passe-bas au niveau du micro de l'ordinateur.

Le profil général du gain du système est aussi cohérent avec les simulations effectuées précédemment. En effet, le système poutre + couplage entre pistes d'actionnement et de mesure peut être modélisé par 2 fonctions de transfert, la première, celle de la poutre, étant un oscillateur résonant du second ordre, et la seconde, celle du couplage, correspondant à la fonction de transfert d'un passe-bas du second ordre à fréquence de coupure très haute (ceci est une première approximation simplifiée pour prendre en compte le couplage) :



Si l'on regarde plus précisément la courbe autour de la fréquence de résonance, on a :

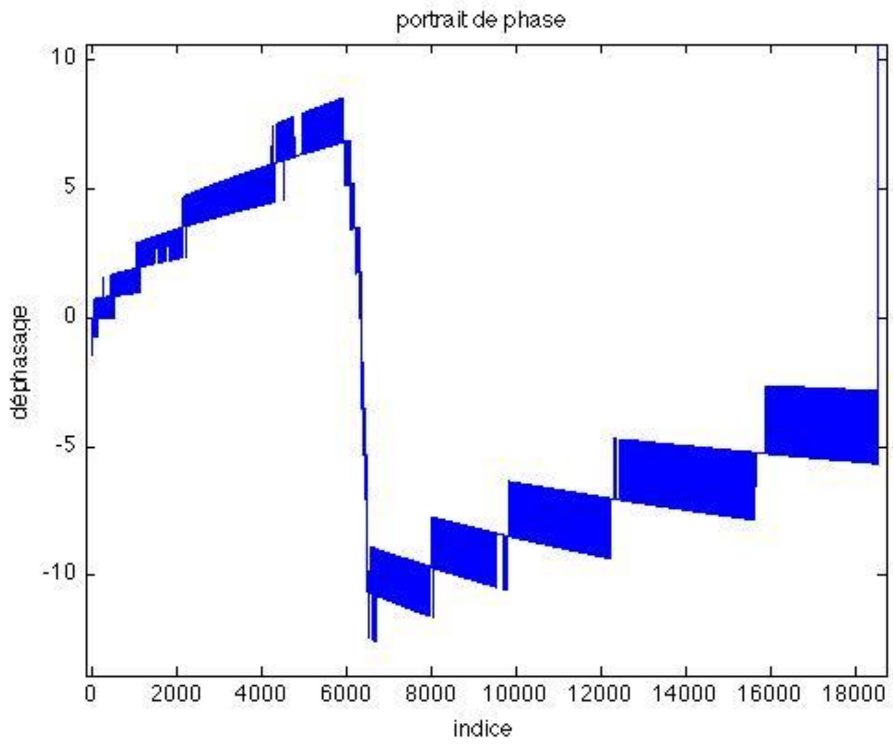


Il est donc ici possible de calculer le facteur de qualité, dont l'expression est Q = $\frac{f_r}{\Delta f}$, avec f_r la fréquence de résonance, et Δf la bande passante à -3dB.

2 - Zerocrossing

Les tests ont ici été effectués avec la même poutre, dans les mêmes conditions. On a réalisé 100 acquisitions.

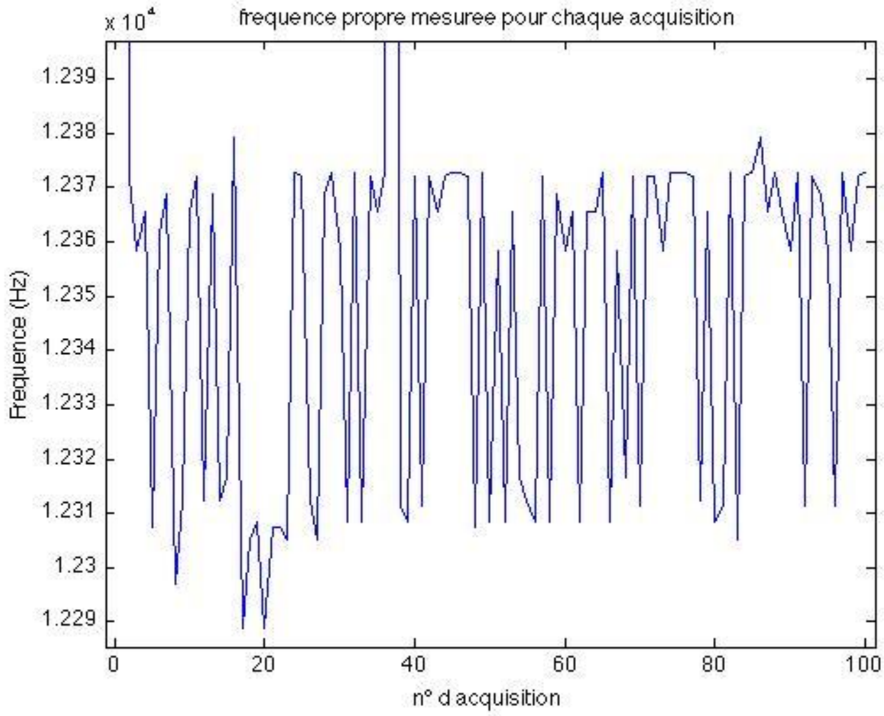
Voici un des portraits de phase obtenus, qui est censé ressembler à la phase d'un système du second ordre.



On observe que, même si la ressemblance avec la phase d'un système du second ordre est assez éloignée (l'axe des ordonnées n'ayant pas réellement de sens physique), la forte pente de "déphasage" peut être observée très facilement, et on peut obtenir la fréquence propre du système.

L'interprétation de cette courbe pose problème, elle ne représente aucune réalité physique. Un autre inconvénient de cette méthode est que, si l'on se trompe d'un écart de 1 point dans les calculs d'indice, l'erreur sur le déphasage peut être énorme, d'autant plus qu'on se situe dans de hautes fréquences.

Voici les différents résultats obtenus pour f_0, pour chaque acquisition :



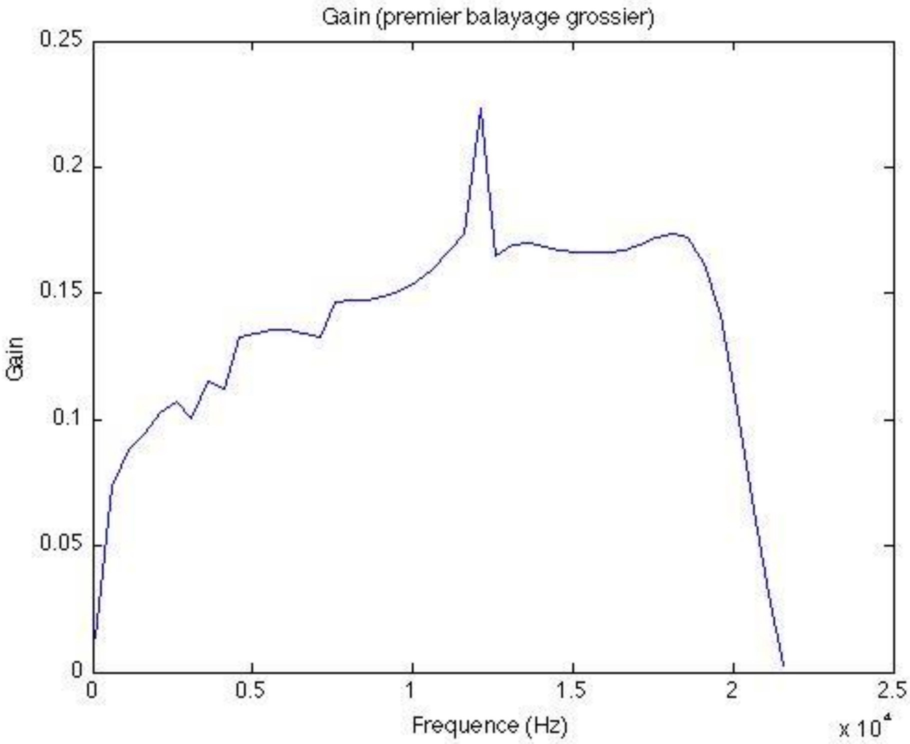
Mis à part quelques points non cohérents, on obtient une fréquence propre assez proche de la fréquence de résonance trouvée précédemment, ce qui est en accord avec la théorie. La largeur de la bande verticale obtenue dans cette figure correspond au pas d'échantillonnage en fréquence autour de 12.3 kHz. La fréquence propre du système mécanique devrait correspondre à la valeur moyenne soit 12 237 kHz.

En récupérant la fréquence de résonance trouvée dans la mesure précédente, et avec les valeurs de fréquence propre trouvées dans cette partie, on peut recalculer le facteur de qualité Q pour vérifier qu'il est en accord avec le Q vu dans le spectre de gain, car $Q = \sqrt{\frac{f_r^2 - f_0^2}{2f_0^2}}$

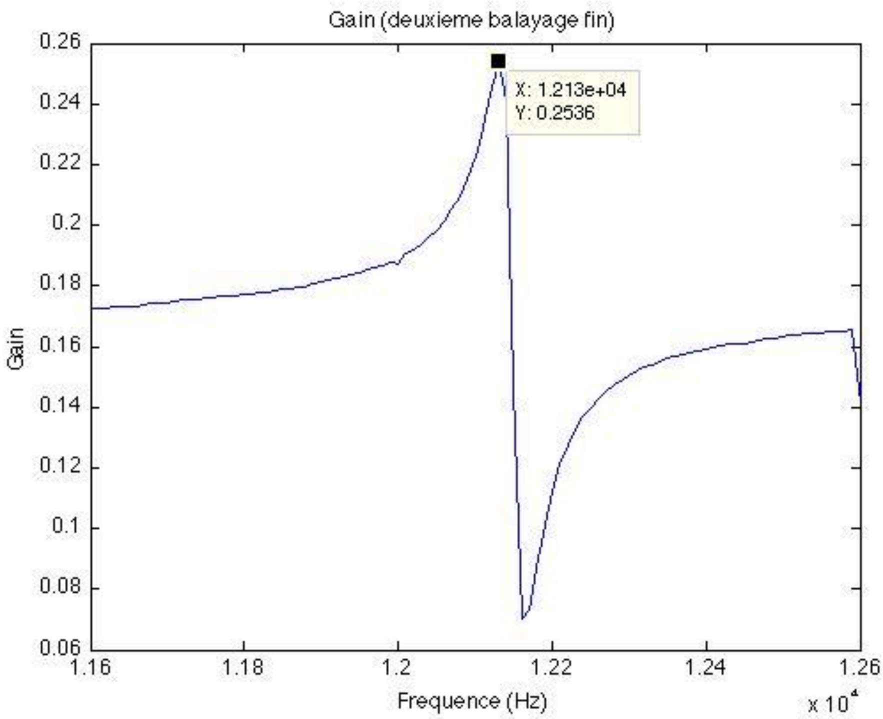
Cependant, les valeurs calculées pour Q nous donnent Q≈ 8, ce qui est incompatible avec le facteur de qualité vu dans la partie précédente.

3 - Mesures de double détection synchrone discrétisée

Pour ces mesures, on utilise toujours une poutre LH dans l'air, vibrant aux alentours de 12 kHz. On effectue tout d'abord un balayage fréquentiel peu résolu pendant 1s:



La figure précédente présente le spectre obtenu par le balayage rapide en détection synchrone. La fréquence maximale est ici bien visible. On refait un balayage, plus précis, autour de cette fréquence, pendant 1s :

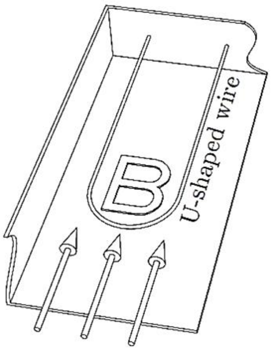


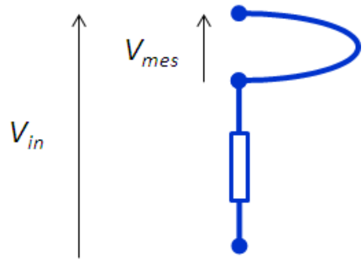
La figure ci-dessus présente l'acquisition réalisée avec une détection affinée autour de la fréquence de résonance. On observe une courbe épurée, peu bruitée et directement exploitable pour le calcul précis de la fréquence de résonance et du facteur de qualité. La courbe est moins bruitée qu'une courbe acquise par la moyenne de dix FFT. Pourtant cette dernière nécessite cinq fois plus de temps d'acquisition.

Cependant, les mesures de phase en détection synchrone ne sont pas aussi bonnes qu'en utilisant la technique « zéro-crossing ». En effet les résultats ne sont pas reproductibles. Pourtant il est théoriquement possible d'obtenir une phase. Cela reste donc un des points à améliorer de ce programme.

4 - Mesures avec un fil en U

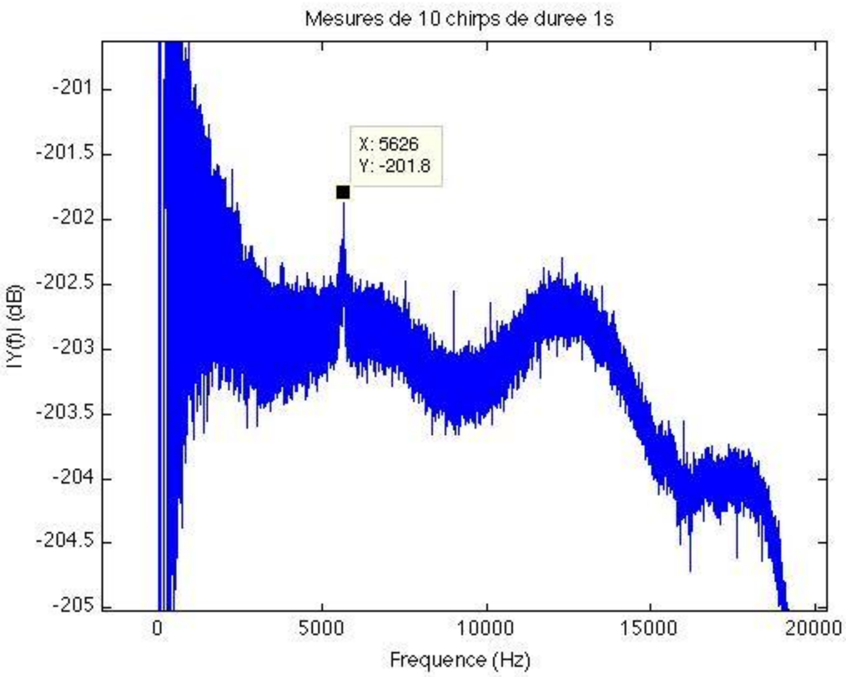
Au-delà du cadre de la micropoutre, un des prototypes de capteur en cours de test est le capteur à fil en U. La micropoutre y est remplacée par un fil conducteur en forme de U également actionné par une force de Laplace.



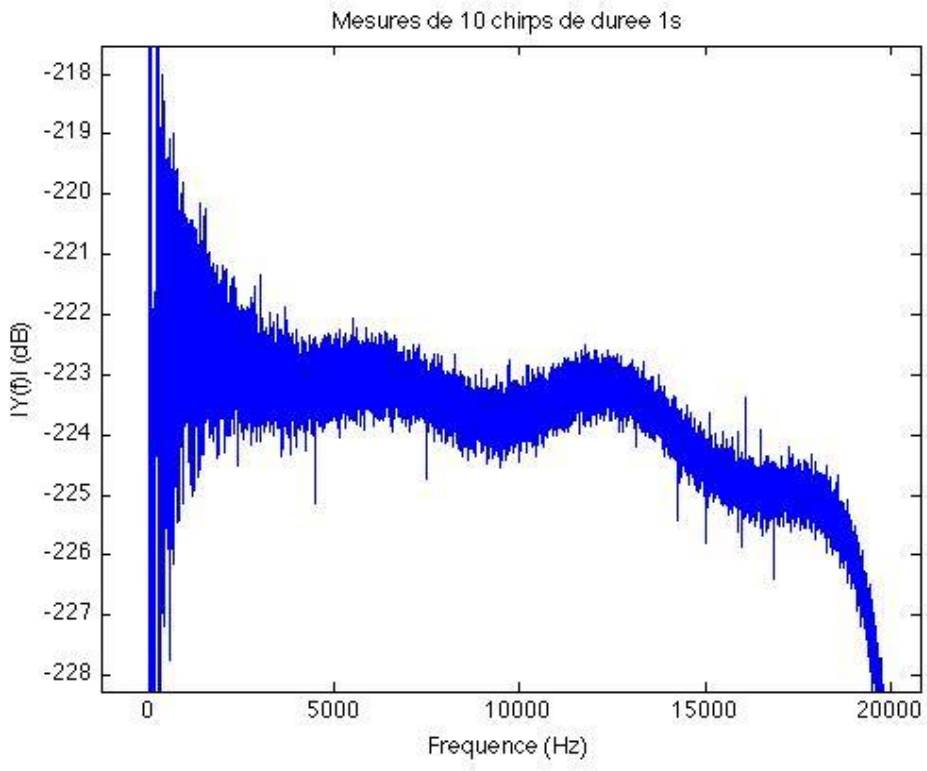


Ce fil est alimenté par une tension sinusoïdale dont la fréquence varie (V_{in}). Il oscille de manière comparable à une micropoutre et se comporte de manière générale comme un oscillateur résonant. La boucle de courant formée par le « U » étant en mouvement dans un champ magnétique, un courant induit circule également dans le fil en « U » en plus du courant d'actionnement. Une faible tension électromotrice (V_{mes}) est donc mesurable aux bornes du fil. Une mesure de déctection synchrone entre V_{in} et V_{mes} caractérise la résonance de la structure à travers le phénomène d'auto-induction.

La figure suivante montre ce que l'on obtient lorsque l'on fait la moyenne sur 10 acquisitions des valeurs absolues des transformées de Fourier du signal de mesure, avec une excitation comprise entre 100 Hz et 22 kHz :



À titre comparatif, la figure ci-dessous montre la même mesure effectuée en court-circuitant le fil en U (mesure de référence), montrant l'absence de résonance mécanique :



Dans le premier graphe, on observe une fréquence de résonance autour de 5,6 kHz, ce qui est cohérent avec les résultats obtenus au vibromètre. La même fréquence est obtenue en faisant diverses mesures, l'expérience est donc reproductible. Ce résultat est prometteur, car pour l'instant les mesures ont été effectuées sans amplification différentielle du signal.

5 - Réponse à un échelon

Une autre manière pour caractériser un système du second ordre est d'étudier sa réponse temporelle à un échelon, ou réponse indicielle.

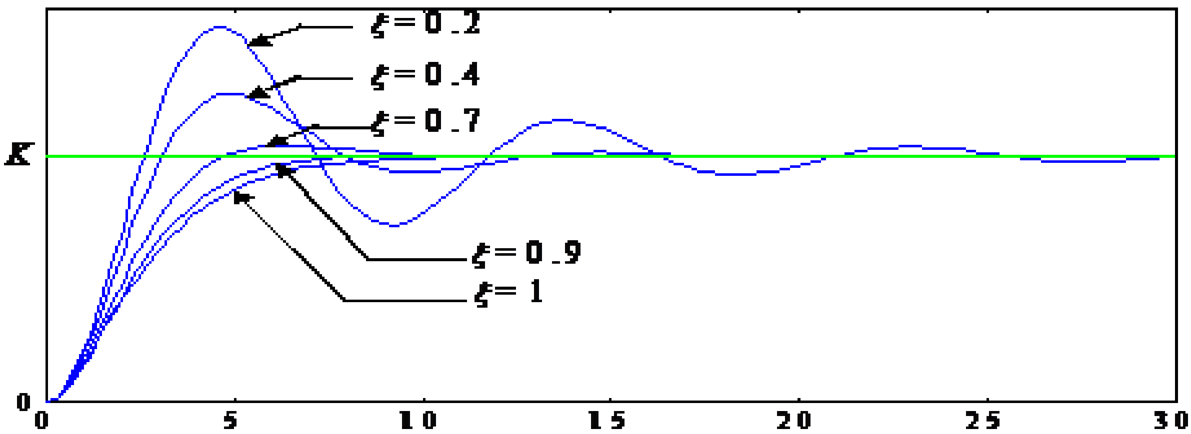


Figure 16 : réponse d'un système du second ordre à un échelon K

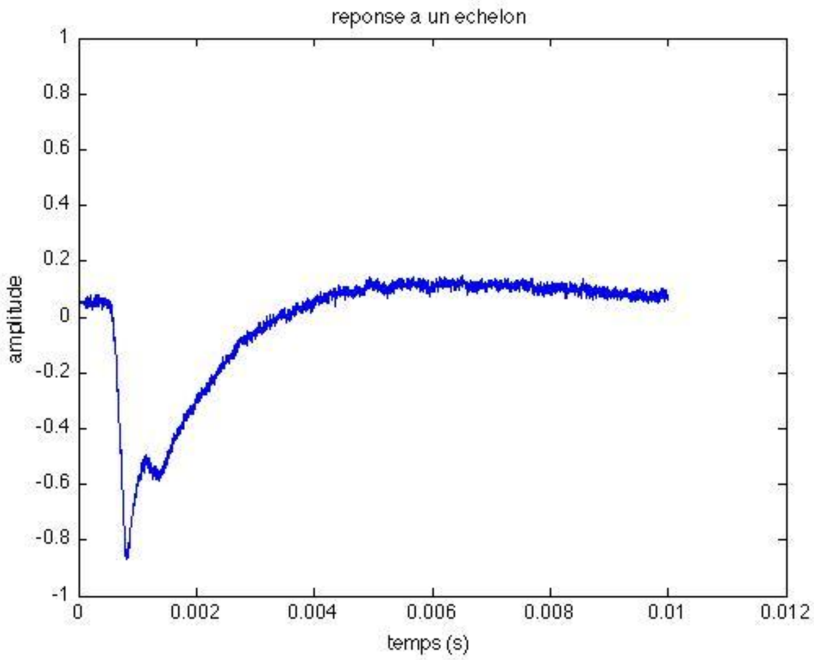
L'amortissement ξ peut être retrouvé grâce à l'expression de $D_{\%}$, la valeur du premier dépassement en pourcentage par rapport à K :

$$D_{\%} = e^{\frac{-\pi\xi}{\sqrt{1-\xi^2}}}$$

Et le temps de réponse à n% permet de retrouver la pulsation de résonance :

$$t_{n\%} = \frac{1}{\xi\omega_n} ln(\frac{100}{n})$$

Après avoir reçu le matériel permettant d'obtenir la voie de référence, je me suis rendu compte qu'une étude de la réponse à un échelon de la poutre était impossible dans les conditions actuelles. Voici ce que j'ai trouvé, en envoyant un échelon en sortie du haut-parleur et en rebouclant aussitôt sur le microphone :



Alors qu'on devrait retrouver un échelon en sortie, on se retrouve ici avec la réponse indicielle d'un système du second ordre. Après calculs, il s'agit en fait d'un filtre passe-bas du second ordre avec fréquence de coupure à 6 Hz environ, qui est placé au niveau de l'entrée du microphone. L'étude de la réponse indicielle n'a donc pas été possible en utilisant la carte son.

V - Développements futurs

Comme nous avons pu le voir au cours de cet exposé, l'actionnement et l'acquisition de la résonance mécanique à l'aide d'une carte son a été réalisé avec diverses méthodes. Néanmoins certaines techniques sont perfectibles ; d'autres pistes peuvent être explorées.

La première piste d'amélioration concerne la technique du « zero-crossing ». Pour avoir des mesures de phases à moins de 5 Hz de résolution, l'augmentation de la fréquence d'échantillonnage est requise. Un échantillonnage à 96 ou 192 kHz avec une carte son externe est envisagé. La résolution verticale de 16 bit peut également être augmentée à 24 ou 32 bits.

L'étude de la réponse à un échelon, rentrant dans le cahier des charges de ce stage, n'a pas pu être effectuée pour des raisons intrinsèques matérielles (à cause d'un filtre passe-haut situé au niveau de l'entrée microphone). Toutefois, cette difficulté pourrait être contournée en utilisant un GBF externe piloté par l'ordinateur.

En utilisant la double détection synchrone, il n'est pas possible pour l'instant d'obtenir une mesure de phase correcte, alors qu'en théorie ce devrait être réalisable. Dans le futur, il serait intéressant d'étudier les différentes sources d'erreur conduisant à ce résultat de phase peu reproductible.

Enfin, en dernier point, une interface graphique, même minimaliste (de type "menu") pourrait être la bienvenue, avec des options de calibration, où l'on puisse par exemple choisir si l'on fait une double détection synchrone, où l'on peut choisir les différents paramètres.

Conclusion

La caractérisation de propriétés viscoélastiques d'un fluide s'effectue via un capteur constitué d'une micro-poutre en silicium immergée dans le fluide. Il s'agit alors de mesurer la fréquence de résonance et la fréquence propre du système ou bien le facteur de qualité du résonateur mécanique. La mesure de cette résonance peut être effectuée optiquement ou électriquement. Puis les signaux de mesures sont acquis de manière classique par un analyseur gain-phase. L'objectif de ce stage était de remplacer cet appareil de détection synchrone par une carte son.

Une technique de détection synchrone spécifique a été développée au cours de ce stage contournant les limitations du système carte son. Des résultats satisfaisant de mesure de résonance ont été obtenus. D'autres techniques de traitement du signal ont également été mises en place. (1) La mesure de phase initialement non accessible a été programmée en utilisant la technique du « zero-crossing ». (2) La transformée de Fourier numérique rapide (FFT) a été utilisée comme technique de référence comparée à la détection synchrone mise en place. (3) La mesure de la réponse indicielle a par ailleurs été tentée. Il apparait que la technique de détection synchrone développée permet en deux temps d'obtenir une meilleure mesure du spectre (gain uniquement) qu'en utilisant la FFT. La mesure de phase par « zero-crossing » permet de mesurer la fréquence propre du système mécanique à 50 Hz près. Finalement les options choisies puis développées répondent de manière complémentaire au cahier des charges initial.

Au cours de ces 2 mois de stage, j'ai pu apprendre à me familiariser avec le monde de la recherche, dans un domaine qui m'était totalement étranger : la rhéologie. Ce domaine nécessite des connaissances aussi bien en physique et mécanique des fluides que, comme ici, en automatisme ou traitement du signal, ainsi qu'en électronique et en conception de capteurs.

Le travail effectué m'a permis d'apprendre à travailler en autonomie, à proposer des solutions à un problème donné, et à avoir une démarche de recherche pour savoir trouver la ou les raisons qui font que les mesures pratiques ne se conforment pas exactement aux simulations théoriques. Il m'a aussi appris à mettre en pratique les connaissances de traitement du signal et d'automatisme que j'ai acquises au cours de ma première année de Master.

Je tiens à remercier chaleureusement Étienne LEMAIRE et Benjamin CAILLARD, qui ont su m'encadrer avec pédagogie dans mon stage, me proposer des pistes de réflexions intéressantes et répondre à toutes mes questions.

D'une manière générale, je remercie toutes les personnes que j'ai pu croiser au laboratoire IMS, qui ont toujours su être accueillantes et ouvertes.

Bibliographie

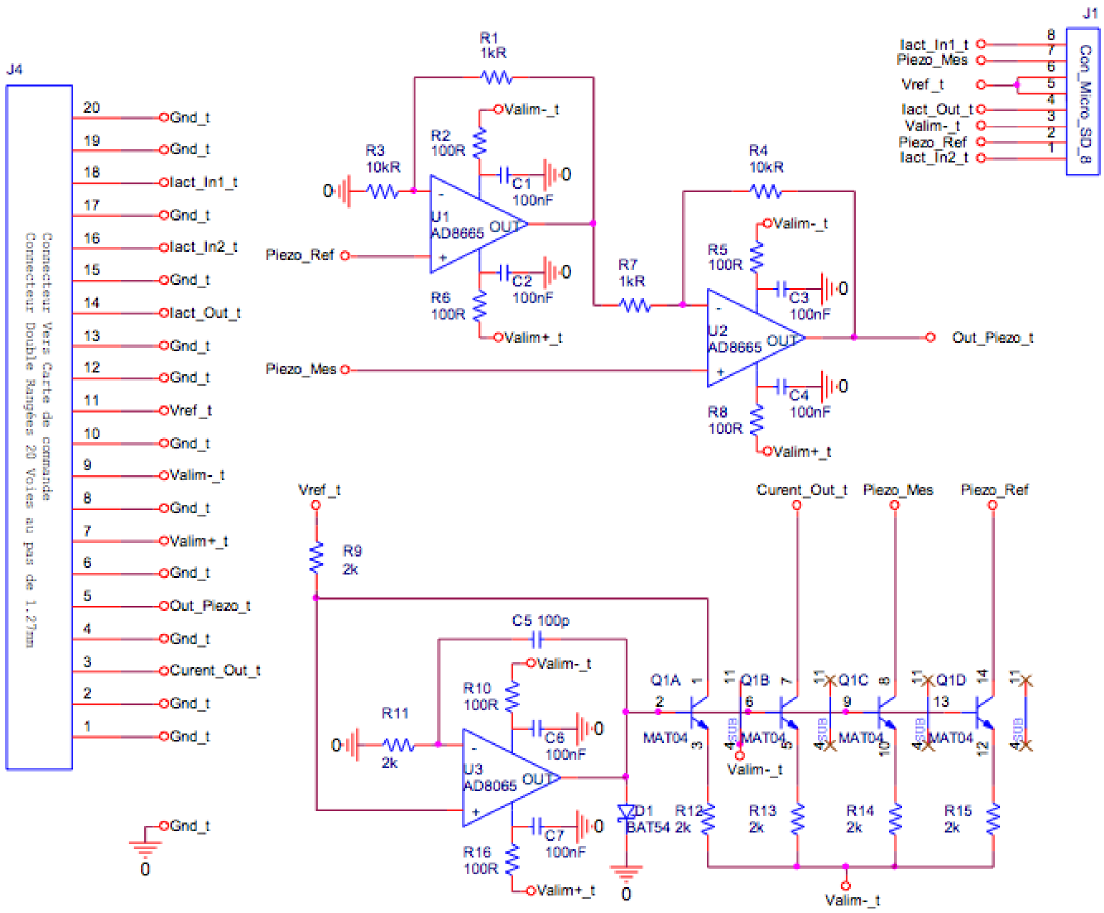
[1] : Synthèse Bibliographique : Actionnement, Détection d'une micropoutre et mesures Rhéologiques. - Étienne LEMAIRE

[2] : Rhéomètre – wikipedia : http://fr.wikipedia.org/wiki/Rh%C3%A9om%C3%A8tre

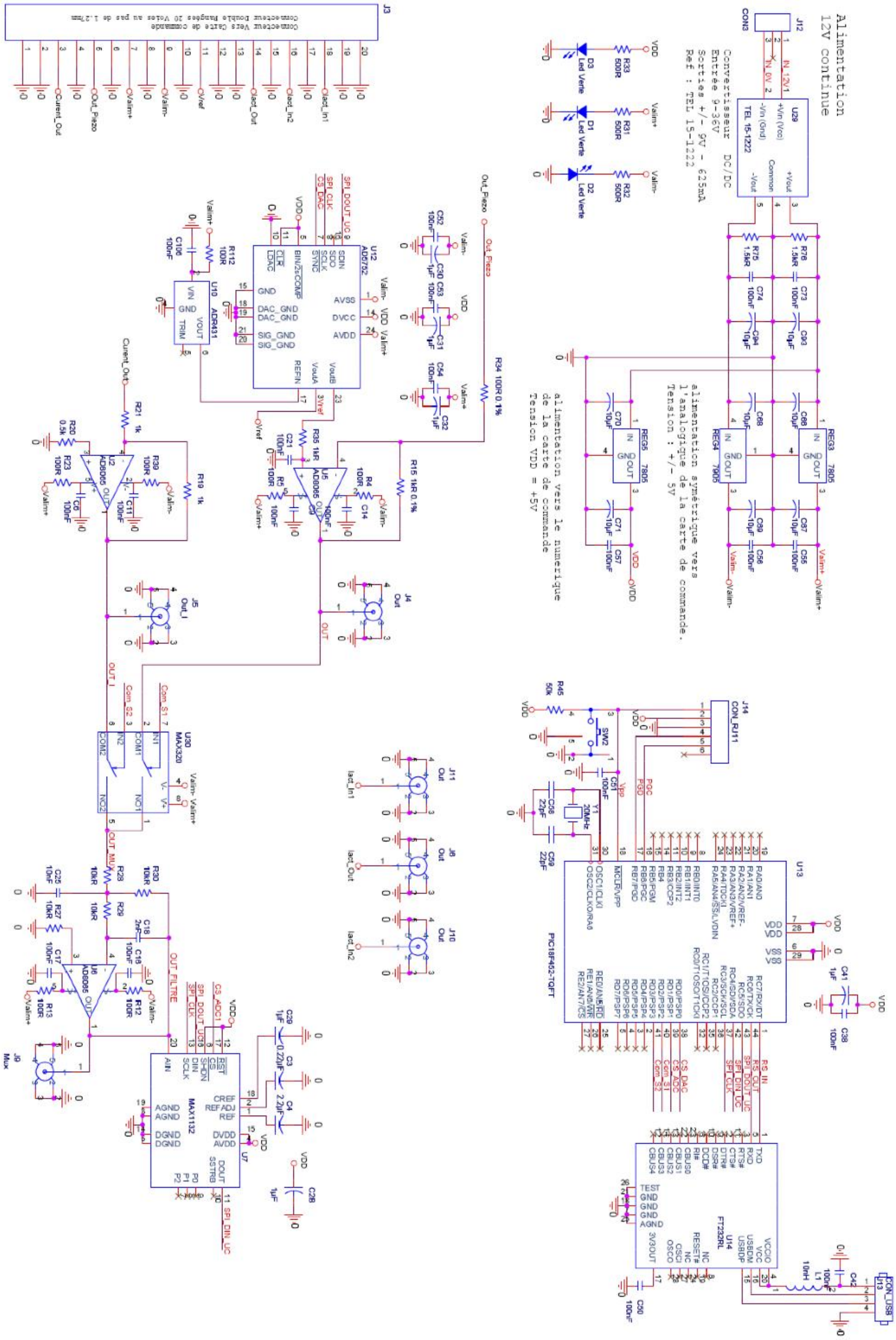
[3] : MICRORHEOMETRE SUR SILICIUM POUR CHIMIE HAUT-DEBIT, Naser BELMILOUD, THÈSE UNIVERSITÉ BORDEAUX I, 2008

[4] : Projet MicRhéo - Système électrique de Détection et de commande - Fabrice MATHIEU - LAAS CNRS

Annexe 1 : schéma de la tête de détection



Annexe 2 : schéma de la carte de commande



Annexe 3 : manuel d'utilisation

Annexe 3 - 1 : main.m

Le programme Matlab final comporte un fichier exécutable main.m, faisant appel aux fonctions qui exécutent les différentes mesures détaillées dans la partie " Méthodes de mesure du gain et de la phase".

On commence tout d'abord par définir les différentes constantes :

```
fs = 44100;              % fréquence d'échantillonnage
nbit = 16;               % nombre de bits
mode = 2;                % 2 = stéréo, 1 = mono
D=1;                     % durée d'acquisition (s)
nb_acq=10;               % nombre d'acquisitions
fchirp = [100, 22000];   % bornes du balayage fréquentiel
bool_phase=0;            % 0 : pas de mesure de phase par zerocrossing
bool_record=0;           % 0 : pas d'enregistrement des données temporelles
```

Puis on fait appel à la fonction moyenne_stereo, qui réalise les mesures de gain par moyenne de transformées de Fourier, et effectue une mesure de la fréquence propre par la méthode de zerocrossing :

```
[f,Gain,fphi,y_tab1,y_tab2]=moyenne_stereo(fs,nbit,mode,D,nb_acq,fchirp,bool_phase,
                                            bool_record);
```

Si bool_record est à 1, les mesures temporelles de la voie 1 et 2 seront enregistrées dans y_tab1 et y_tab2. Il est déconseillé d'enregistrer les mesures temporelles si l'on compte réaliser plus de 500 acquisitions (problèmes de mémoire au niveau de Matlab)

Si bool_phase est à 1, la fonction réalise des mesures de fréquence propre par la méthode de zerocrossing. Cette méthode étant assez gourmande en temps de calcul, si l'on veut effectuer une mesure rapide du gain il est conseillé de la désactiver.

Cette fonction retourne le vecteur fréquentiel f, le gain, un tableau des fréquences propres mesurées pour chaque échantillon, et les mesures temporelles.

La deuxième fonction, ddetec_discret, permet de réaliser une double détection synchrone discrétisée.

```
[F,ampl,fmax, deph, F1, ampl1, deph1]=ddetec_discret(fs,nbit,D,fchirp);
```

Annexe 3 - 2 : moyenne_stereo.m

```
% enregistre nb_acq acquisitions sur les 2 voies.
% réalise la FFT des 2 voies, en fait la moyenne sur les nb_acq
% acquisitions
% affiche la FFT de la voie 2 au fur et à mesure des acquisitions
function
[f,Gain,fphi,y_tab1,y_tab2]=moyenne_stereo(fs,nbit,mode,D,nb_acq,...
                                    fchirp,bool_phase, bool_record)

% ENTREE
% fs            fréquence d'échantillonnage, valeur typique : 44100
% nbit          nombre de bits par échantillon, typique = 16
% mode          pour dire stéréo/mono, 1 pour mono, 2 pour stéréo
% D             durée d'enregistrement pour 1 acquisition (s), typique : 1
% nb_acq        nombre d'acquisitions
% fchirp        bornes du balayage fréquentiel, typique [100, 20000]
% bool_phase    1 si calcul de la phase par zerocrossing
% bool_record   1 si stockage des mesures temporelles dans y_tab1 et y_tab2

% SORTIE
% f             vecteur fréquentiel
% Gain          gain (module de la Tf) des 2 voies
% fphi          fréquences propres mesurées pour chaque acquisition
% y_tab1        mesures temporelles de la voie 1 (référence)
% y_tab2        mesures temporelles de la voie 2 (poutre)
```

On aura besoin, pour faire la transformée de Fourier, du nombre de points de la FFT. La FFT étant plus efficace quand elle est effectuée avec une longueur en puissance de 2. On prendra donc la première puissance de 2 supérieure au nombre de points de l'acquisition (D*fs).

```
% Taille de la FFT (en puissance de 2 pour plus de rapidité =>
% zero-padding)
NFFT = 2^nextpow2(D*fs);
```

Il y a possibilité de charger des données de calibration, qui auront été enregistrées au préalable via la même fonction moyenne_stereo. Cette calibration est le module de la transformée de Fourier d'une référence qu'on choisira.

```
% possibilité de charger des données de calibration
%load calib_poutre
calib = 0;
```

On doit ensuite concevoir notre signal sonore. Celui-ci est un balayage fréquentiel allant de fchirp(1) à fchirp(2), dont la fréquence augmente linéairement. On a pris ici un sinus pour limiter les effets de type "réponse à un échelon" du passe haut à 5 Hz présent dans les haut-parleurs.

```
% Construction du balayage en fréquence
freq=linspace(fchirp(1),fchirp(2),D*fs);
sound=sin(2*pi*cumsum(freq/fs));
```

On assigne un nom de structure aux voies de sortie (via audioplayer) et aux voies d'entrée (via audiorecorder)

```
% assignation
player=audioplayer(sound,fs);
enregistrement=audiorecorder(fs,nbit, mode);
```

Les tableaux de stockage des données temporelles sont initialisés si besoin est.

```
if bool_record==1
```

```
    % initialisation des tableaux de stockage de données voies 1 et 2
    y_tab1 = zeros(D*fs,nb_acq);
    y_tab2 = zeros(D*fs,nb_acq);
else
    y_tab1=0;
    y_tab2=0;
end
```

Il s'agit ensuite de construire le vecteur de fréquence qui nous permettra de tracer la transformée de Fourier.

```
% initialisation fft
Y = zeros(NFFT,2);
% construction du vecteur de fréquences
f = linspace(0,fs/2,NFFT/2);
```

Cette partie du code définit une fenêtre de graphique, et nous permettra plus tard d'afficher le gain au fur et à mesure.

```
%affichage plots en live
figure;
Gain = 20*log(2/NFFT*abs(Y(1:NFFT/2+1)));
ht = plot(f,Gain);
set(ht,'XData',f);
title(['Mesures de ',num2str(nb_acq),' chirps de duree ', num2str(D),'s'])
xlabel('Frequence (Hz)')
ylabel('|Y(f)|')
```

S'il y a mesure de phase, on construit le vecteur temporel :

```
if bool_phase==1
    %Génération des vecteurs fréquentiels et temporels
    freq=linspace(fchirp(1),fchirp(2),D*fs);
    tps = linspace(0,1,D*fs);
    entree=cos(2*pi*cumsum(freq/fs));   %sweep (balayage en fréquence)
else
    phi=0;
end
```

On réalise une boucle for pour décrire chaque acquisition

```
for ind=1:nb_acq
```

Mesures de gain

On commence tout d'abord par jouer le son, puis aussitôt, on enregistre ce qui arrive sur les haut-parleurs :

```
    % joue le son (balayage fréquentiel) et enregistre
    play(player);
    recordblocking(enregistrement, D);
    y=getaudiodata(enregistrement);
```

On calcule le module de la FFT, que l'on ajoute aux modules précédemment calculés pour les autres mesures. Soustraction de la calibration si nécessaire.

```
    % somme des FFT
    Y1 = fft(y,NFFT);
    Y = abs(Y1)+Y-calib;
```

On calcule ensuite le gain moyenné en renormalisant par NFFT. On ne se sert que de la première moitié des données de Y ; en effet, les signaux étant réels, la transformée de Fourier est symétrique en 0.

```
    Gain = 2/NFFT*abs(Y(1:NFFT/2+1,:))/ind;
```

Le code suivant permet d'afficher le gain en temps réel.

```
    % affichage plots en live
    set(ht,'YData',Gain(:,2));
    title(['Mesures de ',num2str(ind),' chirps de duree ', num2str(D),'s'])
    drawnow
```

On stocke les mesures temporelles si besoin est.

```
    if bool_record==1
        % stockage des mesures temporelles
        y_tab1(:,ind) = y(:,1);
        y_tab2(:,ind) = y(:,2);
    end
```

<u>Mesures de la fréquence propre</u>

On commence d'abord par recentrer le signal de sortie en 0, pour pouvoir avoir des mesures de zerocrossing cohérentes.

```
    % calcul de la phase
    if bool_phase==1
        sortie=y(:,2);
        % permet de recentrer le signal en 0
        sortie = sortie - smooth(sortie,5);
```

On récupère les indices de passage à zéro avec la fonction zero_crossing

```
        [inde, te] = zero_crossing(entree,tps);
        [inds, ts] = zero_crossing(sortie,tps);
```

La comparaison des indices des zéros se fait à partir de fstart = 5kHz. Empiriquement, avant cette fréquence, le signal est tellement bruité que les mesures de zerocrossing sont incohérentes.

```
        fstart = 5000;

        % conversion de fstart en indice
        tstart = round((fstart-fchirp(1))*D*fs/(fchirp(2)-fchirp(1)));
```

On supprime donc toutes les mesures correspondant aux fréquences inférieures à fstart, qui sont inutiles.

```
        % suppression des indices antérieurs à celui correspondant à fstart
        inds = inds(inds>tstart);
        inde = inde(inde>tstart);
        l = min(length(inds),length(inde));
        indsdecal = zeros(1,length(inds));
```

Dans cette boucle, on décale les indices de zerocrossing du signal de sortie jusqu'à obtenir un profil de phase ressemblant à la phase théorique.

```
        for Ndecal = 1:200
            indsdecal(1:end-Ndecal+1) = inds(Ndecal:end);
            indsdecal(end-Ndecal+1:end) = 0;
```

Puis on soustrait les indices de zerocrossing d'entrée aux indices de zerocrossing de sortie, pour obtenir Δt, en considérant que le déphasage est à peu près nul pour f=fstart.

```
            indsdecal = indsdecal - indsdecal(1)+inde(1);

            % on considère un déphasage nul à f=fstart
            deph= inde(1:l)-indsdecal(1:l);
```

Δt est ensuite converti en déphasage, en fonction de la fréquence.

```
            % conversion déphasage en nombre de points => déphasage en
            radians
            phi = deph/fs*2*pi.*freq(inde(1:l));
```

La sortie de la boucle de décalage s'effectue via une valeur empirique

```
            % sortie de la boucle de décalage
            if min(phi)>-15 % valeur déterminée de manière empirique
                break;
            end
        end
```

On cherche ensuite la pente négative la plus forte, qui correspond au moment où f=fréquence propre

```
        [v p] = min(phi(101:end-Ndecal-5)-phi(1:end-Ndecal-5-100));

        % conversion indice de dérivée minimum => fréquence
        fphi(ind) = freq(inde(p));
    end
end
```

Enfin, on affiche sur un même graphe le gain mesuré et le gain de la référence.

```
% affichage des FFT voies 1 et 2
hold on;
plot(f,2/NFFT*abs(Y(1:NFFT/2+1,1))/nb_acq, 'g')
```

Annexe 3 - 3 : ddetec_discret.m

```
function
[F,ampl,deph,F1,ampl1,deph1,fmax1]=ddetec_discret(fs,nbit,D,fchirp)
% réalise un premier balayage fréquentiel grossier (changer N pour changer
% la finesse)
% réalise un deuxième balayage fréquentiel autour de la fréquence max de
% gain du premier (changer N2 pour la finesse et delta_f pour la plage)

% ENTREE
% fs        fréquence d'échantillonnage (Hz)
% nbit      nombre de bits par échantillon
% D         durée d'enregistrement pour 1 acquisition (s)
% fchirp    bornes du balayage fréquentiel

% SORTIE
% F         vecteur fréquentiel (balayage grossier)
% ampl      gain (balayage grossier)
% deph      "phase" (balayage grossier)
% F         vecteur fréquentiel (balayage précis)
% ampl      gain (balayage précis)
% deph      "phase" (balayage précis)
% fmax      fréquence de résonance
```

Les constantes suivantes définissent les pas des deux balayages, et l'amplitude du deuxième balayage (fin). Il sera sans doute nécessaire de les modifier en fonction du type de milieu où l'on fait les mesures, ainsi que du type de poutre.

```
N=500;                  % pas fréquentiel pour le premier balayage
delta_f=1000;           % écart de fréquence autour de fmax (2e balayage)
N2=10;                  % pas fréquentiel pour le balayage précis
```

On construit un tableau de fréquences comprises entre les bornes du balayage définies dans main.m, avec un pas de N.

```
% construction du tableau de fréquences grossières
F = fchirp(1):N:fchirp(2);
```

On réalise une double détection synchrone discrète sur toute la gamme de fréquence, ce qui nous donne en sortie une fréquence où l'amplitude est maximale (fmax).

```
[ampl, deph, fmax] = func_discret(fs, nbit, mode, D, F);
```

On effectue maintenant une deuxième double détection synchrone discrète, plus précise, avec un balayage autour du fmax trouvé plus haut.

```
% Deuxième partie : mesures autour de fmax
F1 = max(fmax-delta_f/2,1):N2:fmax+delta_f/2;
D1 = 10;
[ampl1, deph1, fmax1] = func_discret(fs, nbit, mode, D1, F1);
```

Annexe 3 - 4 : func_discret.m

```
function [ampl, deph, fmax] = func_discret(fs, nbit, mode, D, A, F)
%fs          % fréquence d'échantillonnage (Hz)
%nbit        % nombre de bits par échantillon
%mode        % pour dire stéreo/mono
%D           % durée d'enregistrement pour 1 acquisition (s)
%A           % Amplitude du sweep pour éviter saturation de la ref
%F           % Tableau des fréquences d'excitation
nc = 20;     % coupure pour éviter les effets de bord
```

Le code suivant permet de construire un vecteur sweepcos, contenant un cosinus à la fréquence f1, puis un cosinus à la fréquence f2, etc... Chaque cosinus a une durée identique Dd.

```
Dd = D/length(F);
tps= linspace(0,D,D*fs);
sweepcos = zeros(1,D*fs);

for ind = 0:(length(F)-1)
    sweepcos(ind*floor(Dd*fs)+1:(ind+1)*floor(Dd*fs)) =
cos(2*pi*F(ind+1)*tps(ind*floor(Dd*fs)+1:(ind+1)*floor(Dd*fs)));
    % insertion de zéros entre 2 fréquences
%     sweepcos((ind+1)*floor(Dd*fs)-200:(ind+1)*floor(Dd*fs)) = 0;
end
```

De même que dans les programmes précédents, on joue le signal sonore en entier, puis on récupère en voie 1 la référence, et en voie 2 la mesure.

```
playercos=audioplayer(sweepcos,fs);
enregistrement=audiorecorder(fs,nbit, mode);

ampl = zeros(length(F),1);
deph = zeros(length(F),1);

% joue le son en cos et récupère les mesures
play(playercos);
recordblocking(enregistrement, D);
ycos=getaudiodata(enregistrement);
```

```
% Matlab n'enregistre pas aussitôt, du coup l'enregistrement commence à un
% indice variable (entre 300 et 600 échantillons)
% => nécessité de détecter quand ça commence (seuillage à 0.2 pour le cos)
% et décalage des mesures obtenues
for ind1 = 1:D*fs
    if abs(ycos(ind1,1))>0.2
        ycos(1:end-ind1+1,:) = ycos(ind1:end,:);

        % mise à 0 des valeurs finales (proche de 0 pour éviter les
        % divisions par 0)
        ycos(end-ind1:end,:) = 10e-16;
        break;
    end
end
```

Pour chaque fréquence, on effectue une double détection synchrone. Chaque détection synchrone se fait donc sur une durée Dd.

```
for ind3 = 0:(length(F)-1)
```

```
    %récupération des mesures pour une fréquence
    y1 = ycos(ind3*floor(Dd*fs)+1:(ind3+1)*floor(Dd*fs),2);

    %récupération de la ref pour une fréquence
    ycos1 = ycos(ind3*floor(Dd*fs)+1:(ind3+1)*floor(Dd*fs),1);
```

On recrée le signal de sinus en décalant de π/2 le cosinus mesuré en référence.

```
    % création du "sinus" en décalant
    decal = round(1/(4*F(ind3+1))*fs);
    ysin1 = zeros(length(y1),1);
    ysin1(1:end-decal+1) = ycos1(decal:end,1);
```

On effectue la première modulation sur les deux branches : le signal est multiplié d'un côté par un cosinus, de l'autre par un sinus.

```
    modul1 = y1.*ycos1;
    modul2 = y1.*ysin1;
```

Sur les deux branches, on filtre par un passe-bas de type Moving Average.

```
    % Filtre MA
    order = ceil(2*fs/F(ind3+1));
    modul1 = smooth(modul1,order);
    modul2 = smooth(modul2,order);
```

Le gain et la phase sont calculés. Ici, on tronque au début et à la fin de nc points, pour éviter l'influence de la réponse indicielle. nc est une constante qu'on sera peut-être amené à changer en fonction des mesures réalisées.

```
    ampl(ind3+1) = mean(2*sqrt(modul1(1+nc:end-nc).^2 + modul2(1+nc:end-
nc).^2)/A);
    deph(ind3+1) = atan(mean(modul2(1+nc:end-nc)./modul1(1+nc:end-nc)));
end
```

On récupère enfin la fréquence pour laquelle le gain est maximal, qui est notre fréquence de résonance.

```
[a b] = max(ampl);
fmax = F(b);
```